Helwan University – Faculty of Computers and Artificial Intelligence (FCAI) – **Computer Science Department**
CS112 Programming Languages 1

Exercises - Problems Sheet # 6:
Pointers, Strings, & Unions

Spring 2024

No. Of Questions: 14 **No. Of Pages: 4**

Answer the following:

1) Write a short program that declares an integer variable and initializes its value to 25. Then declare a pointer to that variable. Print the value by using that pointer.

2) Write a program similar to the previous one, but instead of an integer, use a string initialized with your own name.

3) Write a program that:

- For a trip, prompts the user for the number of driven miles, and the fuel consumption.
- Calls a function with three parameters: (1) the number of miles, (2) the fuel consumption, and (3) the fuel consumption per mile - which is calculated within the function.
- Prints the fuel consumption per mile – which is calculated within the function.

4) Write a program that reads 5 integers to an array. The integers should then be printed. Use pointer arithmetic.

5) Modify the preceding program so that it would calculate the sum of the integers. Use a pointer variable for the sum.

6) Write a function which takes as arguments: (1) a pointer to an integer array, and (2) the number of items of the array. The function then finds and return the greatest item. In the main function, the user should enter 8 numbers to be stored in the array. The function is then called, and the returned greatest item is printed (in the main function).

7) Write a recursive function that returns the length of a string.

8) Write a recursive function that returns the number of spaces in a string.

9) Write a recursive function that checks if a string is entirely consisting of digits, or not.

10) Write a program that asks the user to enter his/her name and age. Then asks the user whether (s)he is a U.S. citizen. If the answer is Yes, ask the user to enter the name of the state where (s)he comes from. Otherwise, ask the user to enter the name of the country (s)he comes from. (You're required to use a struct and a union in your program).

11) **[From the textbook "C How to program": 10.21]** In this exercise, you’ll design a "starter" HealthProfile structure for a person. The structure’s members should include: the person’s first name, last name, gender, date of birth *(consisting of separate attributes for the month, day and year of birth)*, height, and weight. Your program should have a function that:

- Receives this data and uses it to set the members of a HealthProfile variable.
- Calculates and returns the user’s age in years.
- Calculates the Body-Mass-Index (BMI). ***(BMI = weight / (height * height))***
- **Note:** The program should prompt for the person’s information, create a HealthProfile variable for that person, and display the information from that variable — *including the person’s first name, last name, gender, date of birth, height and weight*, then it should calculate and display the person’s age in years, and his/her BMI.

12) Trace the following:

A

```
void swap_nums( int *x, int *y ) {
  int tmp;
  tmp = *x;
  *x = *y;
  *y = tmp;
}

void swap_pointers( char *x, char *y ) {
  char *tmp;
  tmp = x;
  x = y;
  y = tmp;
}

int main() {
  int a,b; char *s1,*s2;
  a = 3; b=4;
  swap_nums( &a, &b );
  printf( "a is %d\n", a);
  printf( "b is %d\n", b);
  s1 = "I should print second";
  s2 = "I should print first";
  swap_pointers( s1, s2 );
  printf( "s1 is %s\n", s1 );
  printf( "s2 is %s\n", s2 );
  return 0;
}
```

B

```
#include <stdio.h>
int add_two( int x, int y )
{
  static int counter = 1;
  printf("This is the function call of %d,\n",
counter++);
  return ( x + y );
}

/* the main function */
main()
{
  int i, j;
  for ( i = 0, j = 5; i < 5; i++, j-- )
  printf( "the addition of %d and %d is
%d.\n\n", i, j, add_two( i, j ) );
  return 0;
}
```

13) Show the output of the following program:

```
main () {
```

```
union u {
double x;
int y;
} a_union;
struct s {
double x;
int  y;
} a_struct;
printf("The size of  a_union:  %d-byte\n",   sizeof(a_union));
printf("The size of a_struct: %d-byte\n", sizeof(a_struct));
}
```

14) Trace the following program:

```
#define FLOAT_TYPE 1
#define CHAR_TYPE 2
#define INT_TYPE 3

struct var_type {
    int type_in_union;
    union {
        float un_float;
        char un_char;
        int un_int;
    } vt_un;
} var_type;

void print_vt( void ) {
    switch( var_type.type_in_union ) {
        default:
            printf("Unknown type in union\n");
            break;
        case FLOAT_TYPE:
            printf("%f\n", var_type.vt_un.un_float);
            break;
        case CHAR_TYPE:
            printf("%c\n", var_type.vt_un.un_char);
            break;
        case INT_TYPE:
            printf("%d\n", var_type.vt_un.un_int);
            break;
    }
}

main() {
     var_type.type_in_union = FLOAT_TYPE;
     var_type.vt_un.un_float = 3.5;
     print_vt();
     var_type.type_in_union = CHAR_TYPE;
     var_type.vt_un.un_char = 'a';
     print_vt();
}
```

With our best wishes;